ÖREBRO UNIVERSITET

INSTITUTIONEN FÖR
NATURVETENSKAP OCH TEKNIK

Inbyggda System, DT511G

Inbyggda System, teori

A001
4,5 högskolepoäng

Skriftlig tentamen

2023-05-08

Inbyggda system, DT511G (A001)

Tillåtna hjälpmedel: penna, radergummi, engelska-svenska ordbok

Instruktioner:

Läs igenom alla frågor noga.
Ange tentamenskoden på svarsdokumentet.
Du kan svara på *Svenska* eller *Engelska*.
Skriv tydligt **(gäller även för en digital tentamen).**
Detta är en individuell examination - alla misstankar om otillåtet samarbete kommer att rapporteras.
Ansvarig lärare finns tillgänglig via telefon fr.o.m. andra skrivtimmen.
Skriv läsligt!
förklara och motivera era svar

Ansvarig lärare: Pascal Rebreyend, tel: 0702001422

För betyg G krävs 50% av total poäng (20 på 40)

(26 poång gav betyg 4, och 32 poäng gav betyg 5)

Lycka till!

Question 1: (2 points)

You are writing a FreeRTOS program and you are writing the code for an interrupt. Can you use the function xSemaphoreTake in the code of the interrupt?

Question 2: (5 points)

We have a set of aperiodic tasks (see table below). All tasks are independant (except they share the same CPU). Try to find a schedule using the earliest due date (EDD) algorithm. (3 points)

If you are not able to find a feasible schedule for a given set of tasks using the EDD algorithm, will you try another algorithm? (2 points)

Task	Deadline	Execution time
A	5	3
B	12	1
C	9	1
D	11	2
E	3	1
F	7	3

Question 3: (6 points)

We have to find if a set of periodic tasks is schedulable.

The tasks are the following:

Task	Execution time(e_i)	Period (p_i)
A	3	14
B	3	19
C	1	27
D	2	11
E	4	17

- Calculate the maximum response time for all the tasks. (Priorities are set according to RMS) (4 points)
- Do you think it was a good or bad idea to start to compute the maximm response time? (2 points)

  Note: The following equation is used for calculation of maximum response times:

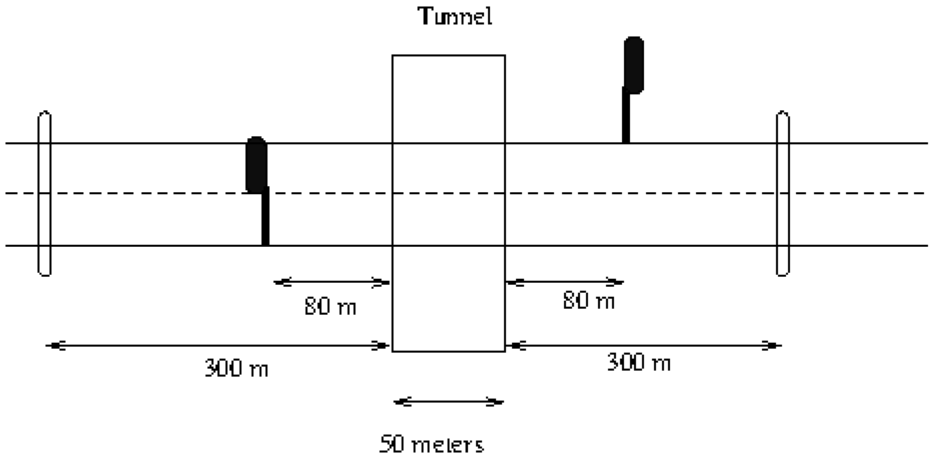
$$R_i = e_i + \sum_{\tau_j \in H_i} \left\lceil \frac{R_i}{p_j} \right\rceil e_j$$

Question 4: (9 points) (Grangesberg Tunnel)

The main road (road 50) going through Grangesberg (a small city north of Örebro) is going through a tunnel. Du to the rounded shape of the tunnel, high vehicules such as trucks and busses can only drive through this tunnel in the middle and therefore they should not meet any other vehicule. Two regulars car can meet in the tunnel without any problem.

Thus, to ensure that high vehicule will no meet any other vehicle in the tunnel, a set of warning lights has been implemented: On each side of the tunnel a light will blink if (and only if) a high vehicle is coming from the other direction. Then vehicules will wait before entering the tunnel until the light is turned off.

Here the schema:



The system work as following:

- Two sensors (300 meters from each side of the tunnel): They work by using a small laser emitted on one side and received on the other side of the road. If the light is not received the input pin Pa or Pb is set to 1, (two pins since we have two different sensors)
- Two lights connected to pins Ba and Bb. (Ba is the light on the same side of the tunnel as the sensor connected to Pa, same for Bb and Pb). Lights are located 80 meters before the entrance of the tunnel.
- We assume that vehicles drives at max 60 km/h where sensors are located.
- If a high vehicle is passing one sensors, the light on the other side of the tunnel should start to blink in less than 60ms. The blinking frequency should be 2 hertz
- The light should stop blinking after a time defined as:
    - time for the vehicule to drive to the blinkling light. We assume the vehicule will drive at 30 km/h between sensors and lights and 15 km/h between the 2 lights
    - If the vehicule who triggers the sensors has to wait due a blinking light, the waiting time and 3 seconds will be added to the driving time explained above.

Distance (m)	50	80	300	0.1
Speed (km/h)	15	15	15	15
Time (s)	12	19	72	0.024

Questions:

1. Since we don’t want to turn on the signal for antenna, what is the minimum amount of time the signal should be at 1 (I.e blocked by a vehicule) if we assume a vehicule driving at 60 km/h and antenna being max 10cm long? (2 points)
2. Describe and write the general structure of the freertos code which can operate such systems. (4 points)
3. Do you see some weaknesses or problems with this system? (2 points)
4. If yes, can you address them and how (For the hardware, you are only allowed to add 2 sensors which are the technically the same as the 2 existing one) ? (3 points)

Question 5: (9 points)

We want to write a code for a system managing a real complex crossing between 2 important and busy roads with traffic lights. The point of the system is to ensure the security by avoiding cars to be block in the middle of the crossing. Another system is used to decide when and which light to turn on or off.

The system is designed in such way:

For every single line entering the crossing, a sensor is activated for every single car passing the line (which is just 1 meter after the traffic light).

For every single line exiting the crossing, another sensor is activated for every car going out of the crossing. Thus, the system may monitor exactly the traffic.

Every sensors is connected to a task or interrupt (*which one you prefer?*). The whole system should trigger an output pin which is 1 if some car(s) are in the crossing (and 0 otherwise). Thus, the system managing the traffic light can delay or not green light in order to have a safe crossing. (*A green light will be only turned on if all cars have been able to exit the crossing*).

Describe the general layout and design of the freertos code. Write one task associated with a sensor for car entering the crossing, one task for a sensor existing the crossing and the task managing the output pin.

(You can freely choose name of global variables, semaphores,...)

Question 6: (3 points)

You are analyzing the following c-code written for an arduino UNO running FreeRTOS and you have the following function:

```
void task1()
{
for (;;)
  {
  button=digitalRead(2);
  if (button==LOW)
    {
      if (state==5)
        {
        state=0;
        analogWrite(11,100);
        delay(500);
        analogWrite(11,0);
        }
      else
        {
          state++;
```

```
          analogWrite(11,255/5*state);
          delay(1500);
          analogwrite(11,0);
        }
     }
  delay(200);
  }
}
```

What do you think about this code?

(Note: The code of this function is fully functional when tested)

Question 7: (3 points)

You are monitoring a realtime system which is composed of the 4 tasks called A,B,C and D.

In the code, the tasks are created with the following priorities: Priority 1 for A, 3 for B and C and 5 for D. (The higher the number is the higher the priority is).

When executing, you see that from time time the OS is increasing the priority of A to 3 or 5 for short period of times. Can you explain what is happening and why?

Question 8: (3 points)

In the case of aperiodic non-preemptive scheduling, two algorithms are commonly used: The Spring Algorithm and the Bratley algorithm. If you have to compare these two different algorithms, how you will proceed? Can you give examples for which the choice between them is easy?